# Minimum Genetic Mutation (View)

A gene string can be represented by an 8-character long string, with choices from `'A'`, `'C'`, `'G'`, and `'T'`.

Suppose we need to investigate a mutation from a gene string `start` to a gene string `end` where one mutation is defined as one single character changed in the gene string.

- For example, `"AACCGGTT" --> "AACCGGTA"` is one mutation.

There is also a gene bank `bank` that records all the valid gene mutations. A gene must be in `bank` to make it a valid gene string.

Given the two gene strings `start` and `end` and the gene bank `bank`, return *the minimum number of mutations needed to mutate from* `start` *to* `end`. If there is no such a mutation, return `-1`.

Note that the starting point is assumed to be valid, so it might not be included in the bank.

**Example 1:**

```
Input: start = "AACCGGTT", end = "AACCGGTA", bank = ["AACCGGTA"]

Output: 1
```

**Example 2:**

```
Input: start = "AACCGGTT", end = "AAACGGTA", bank =
["AACCGGTA","AACCGCTA","AAACGGTA"]

Output: 2
```

**Example 3:**

```
Input: start = "AAAAACCC", end = "AACCCCCC", bank =
["AAAACCCC","AAACCCCC","AACCCCCC"]

Output: 3
```

**Constraints:**

- `start.length == 8`
- `end.length == 8`
- `0 <= bank.length <= 10`
- `bank[i].length == 8`
- `start`, `end`, and `bank[i]` consist of only the characters `['A', 'C', 'G', 'T']`.